CLAIM AMENDMENTS

Claim Amendment Summary

Claims pending

- Before this Amendment: Claims 1-38.

- After this Amendment: Claims 1-15 and 17-38

Canceled claims: 16

Amended claims: 1, 9, 12, 14-15, 17-18, 21-23, 26, 34, and 36-38

New claims: None

This listing of claims will replace all prior versions, and listings, of claims in the Application.

Listing of Claims:

1. **(Currently Amended)** A method comprising:

receiving a policy at a client from a host, the policy including a number of assertions for the client to comply with in order to access one or more resources via the host, and wherein the policy is cached at the client;

determining that the client is complying with at least one assertion;

generating a policy digest at the client for [[a]] the cached policy ~~that applies to a client~~, the policy digest identifying the at least one assertion ~~the client is complying with~~; and

sending a message from the client to the host to access a resource via the host, the message including the policy digest ~~in a request by the client to access a resource~~.

2. **(Original)** The method of claim 1, wherein generating the policy digest includes generating a hash of the cached policy.

3. **(Original)** The method of claim 1, wherein generating the policy digest includes encoding a bit vector identifying selected assertions from the cached policy.

4.   (Original) The method of claim 1, wherein generating the policy digest includes reading an assertion from the policy, assigning a bit value to the assertion, and writing the bit value to a bit vector.

5.   (Original) The method of claim 1, wherein generating the policy digest includes generating a hash of the cached policy if the cached policy is normalized.

6.   (Original) The method of claim 1, further comprising:

incrementing a counter each time the cached policy is used; and

removing the cached policy from a cache at the client when the counter exceeds a limit value.

7.   (Original) The method of claim 1, further comprising:

incrementing a counter for the cached policy when a fault is received at the client in response to using the cached policy; and

removing the cached policy from a cache at the client when the counter exceeds a limit value.

8.   (Original) The method of claim 1, further comprising logging a diagnostic event when a fault is received at the client to identify a system problem.

9.   (Currently Amended) A method comprising:

~~~sending a policy from a host to a client, the policy including a~~~ number of assertions for the client to comply with in order to access one or more resources via the host;

extracting a policy digest from a message received at the host from the client, the policy digest indicating that the client is complying with at least one assertion of the number of assertions;

~~extracting at a host a policy digest identifying a cached policy that applies to a client, the policy digest included in a request to access a resource;~~ and

denying access to the resource if the policy digest identifies an invalid policy.

10.   (Original) The method of claim 9, further comprising issuing a fault for the client if the policy digest identifies an invalid policy.

11.   (Original) The method of claim 9, further comprising decoding the policy digest.

12.   (Currently Amended) The method of claim 9, further comprising decoding a bit vector of the ~~cached~~ policy.

13.   (Original) The method of claim 9, further comprising reading an assertion from the policy digest.

14.   (Currently Amended) The method of claim 9, further comprising reading a row hash of the ~~cached~~ policy.

15.   (Currently Amended) A system comprising:

a processing unit; and

a system memory accessible to the processing unit, the system memory including:

a message processor to:

receive a message from a client to access a resource; and

extract a policy digest from the message, the policy digest indicating that the client is complying with one or more of a number of assertions of a policy in order to access one or more resources via the system and the policy digest including a bit vector identifying the one or more assertions; and

a fault generator to:

return an invalid digest fault to the client when a length of the bit vector is not valid; and

determine whether the one or more assertions are valid when the length of the bit vector is valid

~~a policy digest identifying at least one cached policy that applies to a client; and~~

a messaging module denying access to a resource if the policy digest identifies an invalid policy for the resource.

16.   (Canceled)

17.   (Currently Amended) The system of claim 15, wherein the ~~messaging module~~ message processor is configured to ~~decodes~~ decode the policy digest.

18.   (Currently Amended) The system of claim 15, wherein the ~~policy digest is a bit vector of a cached policy~~ fault generator is configured to return an invalid policy fault to the client when at least one of the one or more assertions specified in the policy digest is invalid.

19.   (Original) The system of claim 15, wherein the policy digest is a row hash of a normalized policy.

20.   (Original) The system of claim 15, wherein the policy digest identifies at least one selected assertion.

21.   (Currently Amended) A system comprising:

a processor; and

a memory accessible to the processor, the memory including:

a digest generator to:

generate a policy digest based on one or more policies received at a client from a host, the one or more policies each specifying at least one assertion that the client must comply with in order to access a resource via the host; and

place a bit vector in a header of a message to access a particular resource of the host, the bit vector including one bit for each assertion of a particular policy and including one bit for each assertion of an additional policy referenced by the particular policy

~~a policy digest for a cached policy that applies to a client, the policy digest identifying at least one assertion the client is complying with; and~~

~~a messaging module including the policy digest in a request by the client to access a resource.~~

22.   (Currently Amended) The system of claim 21, ~~wherein the~~ further comprising a messaging module to ~~encodes~~ encode the policy digest.

23.   (Currently Amended) The system of claim 21, ~~wherein the policy digest is a bit vector of a cached policy~~ further comprising a cache including the one or more policies.

24.   (Original) The system of claim 21, wherein the policy digest is a row hash of a normalized policy.

25.   (Original) The system of claim 21, wherein the policy digest identifies at least one assertion selected by the client.

26.   (Currently Amended) A computer program product encoding a computer program for executing on a computer system a computer process, the computer process comprising:

    receiving a policy at a client from a host, the policy including a number of assertions for the client to comply with in order to access one or more resources via the host, and wherein the policy is cached at the client;

    determining that the client is complying with at least one assertion; and

    generating a policy digest at the client for [[a]] the cached policy that applies to a client, the policy digest identifying the at least one assertion the client is complying with, and including the policy digest in a request by the client to access a resource.

27.   (Original) The computer program product of claim 26 wherein the computer process further comprises generating a hash of the cached policy.

28. (Original) The computer program product of claim 26 wherein the computer process further comprises encoding a bit vector of the cached policy.

29. (Original) The computer program product of claim 26 wherein the computer process further comprises reading an assertion from the policy, assigning a bit value to the assertion, and writing the bit value to a bit vector.

30. (Original) The computer program product of claim 26 wherein the computer process further comprises generating a row hash of the cached policy if the cached policy is normalized.

31. (Original) The computer program product of claim 26, wherein the computer process further comprises:

incrementing a counter each time the cached policy is used; and

removing the cached policy from a cache at the client when the counter exceeds a limit value.

32. (Original) The computer program product of claim 26 wherein the computer process further comprises:

incrementing a counter for the cached policy when a fault is received at the client in response to using the cached policy; and

removing the cached policy from a cache at the client when the counter exceeds a limit value.

33.   (Original) The computer program product of claim 26 wherein the computer process further comprises triggering a diagnostic event when a fault is received at the client.

34.   (Currently Amended) A computer program product encoding a computer program for executing on a computer system a computer process, the computer process comprising:

extracting at a host a policy digest <u>included in a message from a client</u> ~~identifying a cached policy that applies to a client~~, the policy digest <u>indicating that the client is complying with an assertion required to access a resource via the host and the assertion is associated with a policy</u> ~~included in a request to access a resource~~; and

denying access to the resource if the policy digest identifies an invalid policy.

35.   (Original) The computer program product of claim 34 wherein the computer process further comprises decoding the policy digest.

36.   (Currently Amended) The computer program product of claim 34 wherein the computer process further comprises decoding a bit vector of the ~~cached~~ policy.

37.   (Currently Amended): The computer program product of claim 34 wherein the computer process further comprises reading ~~an~~ <u>the</u> assertion from the policy digest.

38.   (Currently Amended) The computer program product of claim 34 wherein the computer process further comprises reading a row hash of the ~~cached~~ policy if the ~~cached~~ policy is normalized.